

RECONFIGURATION OF SHIPBOARD POWER SYSTEMS

By:

K.R. Davey
R.E. Hebner

1st WSEAS International Conference on Electrosience and Technology for Naval Engineering
and All-Electric Ship, Athens, Greece, July 12-13, 2004

IASME Transactions, vol. 1, no. 2, April 2004, pp. 241-246

PR - 362

Center for Electromechanics
The University of Texas at Austin
PRC, Mail Code R7000
Austin, TX 78712
(512) 471-4496

April 23, 2004

Reconfiguration of Shipboard Power Systems

KENT DAVEY and ROBERT HEBNER

Center for Electromechanics

The University of Texas at Austin

1 University Station R7000, Austin, Texas, 78759

UNITED STATES OF AMERICA

k.davey@mail.utexas.edu, <http://www.utexas.edu/research/cem/>

Abstract: The electrical system on a ship is well contained. Among the challenges proffered by naval systems is real time monitoring for the purpose of fault analysis and reconfiguration. The power system can be considered as a grid of interconnected trunk lines each with its own equivalent parallel load impedance and series transmission impedance. These equivalent impedances provide a natural means of reducing a rather large, complex system to a small compact system. The reduced system can be optimized to maximize power flow through the equivalent parallel impedances and minimize the loss in the series components. Binary integer optimization techniques offer the most promise in solving these problems quickly.

Key-Words: Reconfiguration, optimization, equivalent impedance, circuit

1 Introduction

Butler and Sarma have discussed a method for accomplishing shipboard reconfiguration using a heuristic rule based approach, based on the assumption that the largest loads be re-routed first [1]. The same authors also suggest treating the network as a fixed charge flow problem hierarchically supplying vital loads first [2]. Central to any reconfiguration scheme is the load flow analysis. Lewis recommends a forward-backward sweep analysis which has been shown to be equivalent to multiplying the bus impedance matrix by the current injections [3].

It is useful to represent the system using equivalent pi sections over trunk lines. This condensed system can be computed quickly with voltage and current measurements at branch points. An optimization procedure is applied to a small test system with over 16 million switch configurations. One advantage of this reduction process is that global optimum configurations can be determined, a very difficult task with larger systems using either discrete or continuous variable representations of the switches. An integer optimization procedure is considered which allows local solutions to be determined rapidly.

2 Approach

Three consecutive data points of a signal are sufficient to convert the signal to phasor representation, specifically an expression for the

ratio of the two signals. The equations for this process are listed in the appendix. A single-phase or multiphase load to ground can be represented in terms of an equivalent series and parallel impedance if the voltage and current are known on the feed end(s) to the load. The series and parallel load impedances between nodes 5 and 9 in Fig. 1 are computed as follows:

$$Z_s = \frac{2(V_5 - V_9)}{(I_5 - I_9)} \quad (1)$$

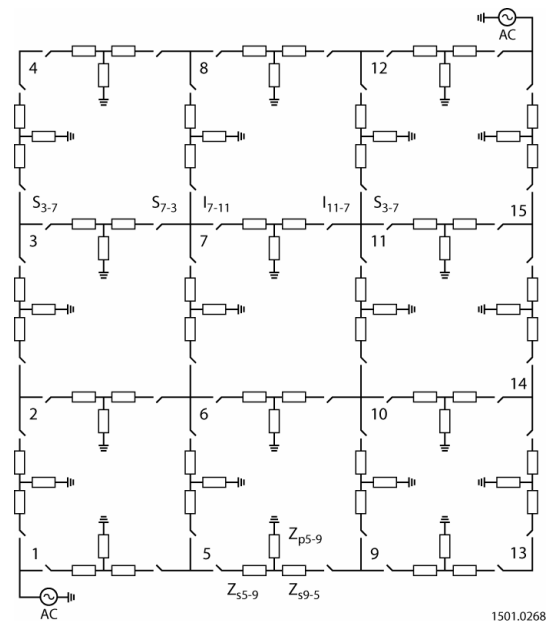


Fig. 1. Sixteen point grid with every trunk line represented by two switches, two series impedances, and one parallel load impedance

$$Z_p = \frac{(V_9 I_5 - V_5 I_9)}{I_5^2 - I_9^2} \quad (2)$$

Generalizing this result as a ratio of two signals yields

$$Z_{sij} = 2 \frac{V_i}{I_i} \left(1 - \frac{V_j}{V_i} \right) / \left(1 - \frac{I_j}{I_i} \right) \quad (3)$$

$$Z_{pij} = \left(\frac{V_j}{I_i} - \frac{V_i}{I_i} \cdot \frac{I_j}{I_i} \right) / \left(1 - \left(\frac{I_j}{I_i} \right)^2 \right) \quad (4)$$

If the values can be updated in real time, the equivalent impedances quantities are dynamic and capable of representing even the most nonlinear system for short time periods, and they are the key to fault control.

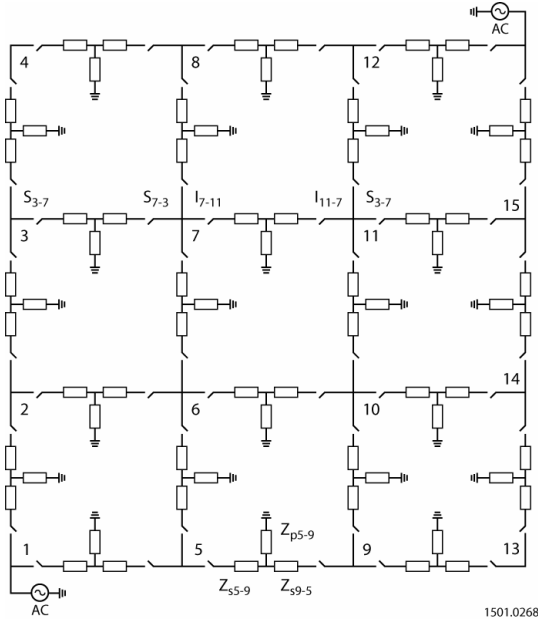


Fig. 1. Sixteen point grid with every trunk line represented by two switches, two series impedances, and one parallel load impedance

3 System reconfiguration

The primary purpose of the reconfiguration algorithm is to find switch settings that maximize power delivery while minimizing transmission loss, subject to the constraint that the current loading on a trunk line remain below its steady state rating. Mathematically, the functional to minimize is

$$\begin{aligned} \mathfrak{S} &= \sum I_s^2 R_s - \sum I_p^2 Z_p \\ \text{subj} \\ I_s &\leq I_{\text{trunk rating}} \\ \text{Gen}_{\text{load}} &\leq P_{\text{rated}} \end{aligned} \quad (5)$$

The last line insures that no generator exceeds its rating. When loads have different weighting priorities w_j , the index changes to

$$\begin{aligned} \mathfrak{S} &= \sum I_s^2 R_s - \sum_{j=1}^m w_j I_{pj}^2 Z_{pj} \\ \text{subj} \\ I_s &\leq I_{\text{trunk rating}} \\ \text{Gen}_{\text{load}} &\leq P_{\text{rated}} \end{aligned} \quad (6)$$

For the 16 point grid shown in Fig. 1, each trunk line and the loads on that line are represented by two series impedances Z_s and one load impedance Z_p . Because dynamic load impedances are used to compute useful power on a trunk line, this type of representation is suitable for fairly complicated systems. The number of points will vary, depending on the size of the system as will the placement of the sources. If the size of the trunk lines is chosen carefully, additional trunk lines from the sources to the center of the grid are unnecessary. Load flow is controlled by the state of the switches. The switches on the ends of any one trunk line are never both open in normal operation.

4 Optimization for Steady State Configuration

The optimization in this section is designated steady state only in the sense to differentiate it from a fault transient. It is designated a discrete variable optimization because the unknowns, i.e., the switch settings, have only two states. The status of the switches will change dynamically throughout the day to reflect new steady state load changes. A voltage nodal analysis is recommended. The number of voltage nodes N_n for an N_{row} by N_{col} grid is

$$N = N_{\text{col}} \cdot (2 \cdot N_{\text{row}} - 1) + N_{\text{row}} (N_{\text{col}} - 1) \quad (7)$$

This formula includes the interstitial nodes created by R_p . The number of switches N_s for this same N_{row} by N_{col} grid is

$$\begin{aligned}
N_s &= 4 + 4 \cdot (N_{\text{col}} - 2)(N_{\text{row}} - 2) \\
&+ 3 \cdot (N_{\text{row}} - 2) + 3 \cdot (N_{\text{col}} - 2) \\
&= 4N_{\text{row}}N_{\text{col}} - 5(N_{\text{row}} + N_{\text{col}}) + 8
\end{aligned} \tag{8}$$

If the analysis were performed blindly, 2^{N_s} analysis iterations would be required for optimization. The number of required calculations is reduced because of the constraints existing in steady state. The switches on both ends of a trunk line cannot both be open.

Islands of groups of trunk lines are generally not allowed. The one exception happens when the generation capacity is exceeded.

The exception to both rules occurs during fault correction. Applying only constraint 0, the number of systems to be analyzed becomes $2^{N_s} - N_s$.

Discrete optimization consists of the following:

1. Assume a switch state for the grid
2. Perform a Kirchoff voltage node analysis using the equivalent impedance representation
3. Check whether the currents in all trunk lines are less than rated capacity
4. Verify that the power and current for any generator remains below its rating level
5. Compute $\sum I_s^2 R_s - \sum I_p^2 Z_p$ for solutions passing the constraints of equation (5).
6. Repeat steps 1 through 5 for all feasible switch states
7. Select the configuration minimizing $\sum I_s^2 R_s - \sum I_p^2 Z_p$

The equations for step 2 are easily programmed through the definition of a single table, one designating the number of neighbors for each node, and the node number of those neighbors.

Table 1 shows how the nodes and their neighbors are identified. Let V_{15} designate the voltage between nodes 1 and 5 across the load impedance Z_{P15} , and in general V_{ij} . Let the switch resistance S in the ij branch, but closest to node 1 be represented as R_{Wij} and have the value of 0 or $10^9 \Omega$, depending on whether the switch is closed or open. The network can be represented in terms of two equations. At node i ,

$$\sum_{j=1}^n \frac{(V_i - V_{ij})}{(R_{Wij} + Z_{Sij})} = 0 \tag{9}$$

The nodal equation at the mid point of the trunk line between nodes i and j is

$$\frac{(V_{ij} - V_i)}{(R_{Wij} + Z_{Sij})} + \frac{(V_{ij} - V_j)}{(R_{Wji} + Z_{Sij})} + \frac{V_{ij}}{Z_{Pij}} = 0 \tag{10}$$

This set of equations is linear, and the governing matrix is sparse.

It should be obvious that whenever one end of a trunk line is open, Z_s and Z_p become combined as a single impedance Z . Real time current and voltage measurements can no longer distinguish between the two entities. How should the minimization index $\sum I_s^2 R_s - \sum I_p^2 Z_p$ be evaluated under this condition? Three options are:

Table 1. Nomenclature used for identifying nodes and neighbors

Node (i)	of Neighbors (n)	Neighborhood node (j)			
1	2	2	5		
5	3	1	6	9	
9	3	5	13	10	
13	2	9	14		
2	3	1	6	3	
6	4	5	10	7	2

1. Continue to use the Z_s from the last known measurement available when both ends were closed
2. Assume that $Z_s = 0$
3. Store a measured or computed trunk line impedance for this condition

Option 2 is a poor choice since it encourages closure of one end of every trunk line. Both options 1 and 3 should be able to handle this condition adequately. In these cases, the impedance Z_p will be that derived from measurement (V_i / I_i), less this estimate.

5 Testing Reconfiguration

Consider the three by three test grid shown in Fig. 2. Work the problem with quantities expressed in per unit. Let $V_1 = V_2 = 1$, $Z_s = 0.1$, and $Z_p = 1$. Listed adjacent to the arrows is the value of current determined by solving equations (9) and (10). All the switches are *closed*.

Lowering to 2 A the maximum current running on trunk line between the source V_1 and node 2 may appear to be a minor perturbation. Fig. 3 shows the optimum constraint with the additional proviso that no trunk line be without power. Since per-unit values are employed, the current through every parallel load should be in

the range of 0.7 to 0.9 A for a 1 pu load. Optimal configurations for tightly coupled systems are not intuitive. The optimization index (equation 5) when only switch 1 or switch 2 on trunk line 1 is switched is -1.36 and -2.41 respectively, whereas with the optimal configuration shown, the index is -2.85. The point to be noted is that a *considerable improvement in power quality with loss reduction can be realized with this approach.*

Is it necessary to add the constraint that power is delivered to every trunk line? Assume trunk line 1 between the voltage source and node 2 is constrained to carry no more than 2 A as above, but that no requirement exists for power delivery to every trunk line. With no power delivery constraint to every trunk line, the new configuration becomes that shown in Fig. 4. The point of the exercise is to confirm that the constraints are required.

6 Expediting the Reconfiguration Algorithm

6.1 Condensing the allowed switch set

Since the objective is to optimize steady state performance, elimination of trunk lines for the purpose of minimizing the index in equation (5)

is not allowed. Inferences commensurate with this objective are that the switches on either side of a trunk line are not both allowed to be open, e.g., switch pairs 2-3 and 3-4, or 2-6 and 6-10 in Fig. 2. This rule would also affect switch settings around corner trunk lines. Switch pairs 2-3 and 4-7 or 2-3 and 7-12, also, may not be open simultaneously. For a 24-switch system such as

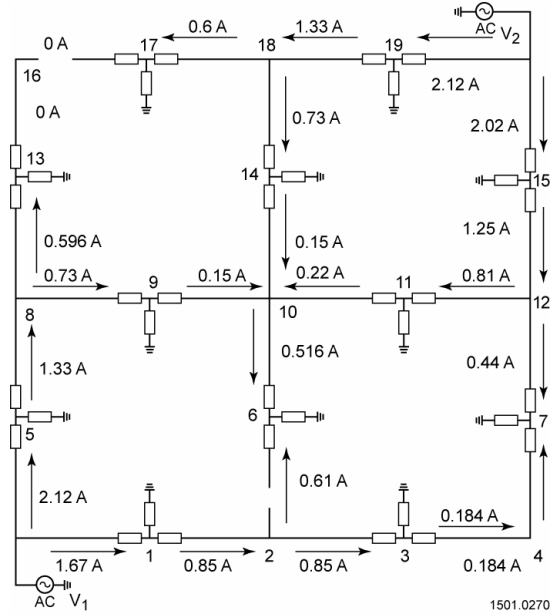


Fig. 3. New configuration resulting from the requirement that trunk line 1 carry no more than 2 A

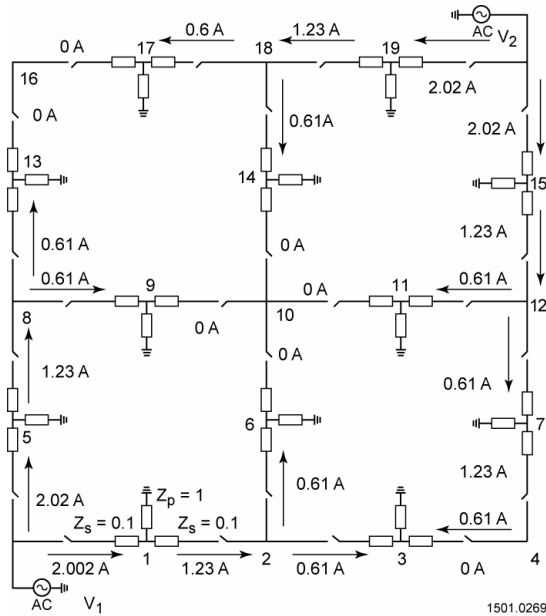


Fig. 2. Test grid for reconfiguration analysis

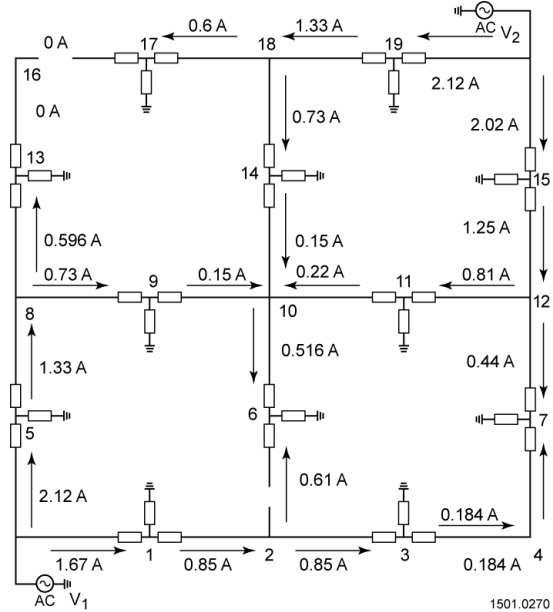


Fig. 4. Configuration resulting from constraining the current between V_1 and node 2 to be less than 2 A, with no constraints that power be delivered to every trunk line

that being examined in this test scenario, the number of switch permutations is reduced from 16,777,216 with no constraints to 217,728 with the constraints imposed. The 217,728 analyses are completed in 56 s on a 3 GHz Pentium IV, complex, single phase. These restrictions must be relaxed if generator or trunk line current ratings are violated.

6.2 Truncating the analysis set

Because of the exponential growth of the number of optimization possibilities, large systems cannot be computed as a unit. Such a technique has the false appeal that for a very large system, switch settings in the upper right quadrant have little effect on power flow in the lower right. The problem is that the neighborhood is always close at the cut points.

In the region highlighted in Fig. 5, Trunk lines 8-13, 10-14, and 12-15 must be cut. What should happen at nodes 8, 10, and 12? Among the options are the following (for node 8):

1. Place a load impedance equal to the voltage divided by the current (V_8 / I_{8-13}) of the last known configuration.
2. Place a source of system voltage V behind that node in series with an impedance equal to $(V_8 - V) / I_{8-13}$.
3. Place a current source at node 8 equal to the current carried by that trunk line for the last known configuration.

Options 1 and 2 will necessitate the use of negative impedances. This inherently leads to the additional constraint that the current through the cut line not exceed the rating of the cut line. Option 3 is safe, but also an approximation. All the options listed above provide a solution different from that obtained analyzing the system in its entirety.

6.3 Continuous variable optimization

If the switches are treated as continuous nonlinear variable resistance in a variable x , the problem can be solved using variable metric optimization. The resistances can be fitted within $0 < x < 1$, and the function $\lambda(x - x^2)$ can be added to the minimization index. λ is a Lagrange multiplier. This trick encourages the function to move toward 1 or zero, values for which the resistance has a value of 1 or a very high value. This implementation in no way insures the location of a global minimum.

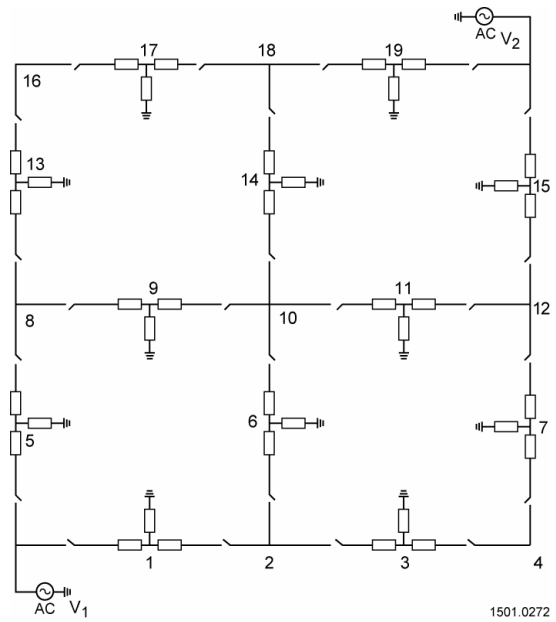


Fig. 5. Reconfiguring only a portion of the total grid

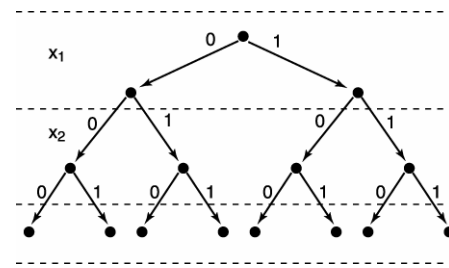


Fig. 6. Branching used in the combinatorial integer programming algorithm proposed for reconfiguration.

6.4 Binary Integer Optimization

A binary integer programming problem takes the form

$$\text{minimize } \vec{f}'\vec{x} \text{ such that } A\vec{x} \leq \vec{b} \quad (11)$$

The unknown x can only have the value 0 or 1. Branch and bound algorithms have been found especially powerful for these types of problems [4,5,6].

The algorithm proceeds by treating the variables x as continuous. Suppose a solution of 0.45 is delivered for x_1 . Two new problems are then generated (referred to as a branch); the first will have the added constraint $x_1 = 0$, while the second branch will have the constraint $x_1 = 1$. The branches result in a decision tree with 2^n possibilities at the bottom of the tree as shown in figure 6. The algorithm has to decide whether to branch or to jump to the next node which is the opposite setting for the existing switch. If the relaxation problem returns an infeasible solution or its optimization index is greater than the

current best integer value, that node is removed, and no other branches are searched below that node. If a better integer solution is found than the best existing integer solution, it updates the current best solution and moves on to the next node. Lastly, if an objective is found to be better than the current best value, but the solution is not integer, then the algorithm branches and proceeds down the tree. Branching is what causes the problem to grow. *The key to this algorithm is that a branch is not formed unless the performance index is shown to improve at a node when the unknown is a non-integer value.*

This algorithm can be performed with great speed if the problem can be linearized about some operating point. The procedure is as follows.

1. Assume a switch configuration x_0 , even one that is not feasible is suitable to start. A binary vector, x has value 0 or 1; each component of the vector represents a switch.
2. Compute the optimization index \mathfrak{S}_0 in eq. (5) and the commensurate line currents I_{s0} .
3. As each switch is toggled sequentially, compute the change in the optimization index $\Delta\mathfrak{S}$ and the line currents ΔI . The objective is to derive an approximate expression for the current change and index change at the present state. Let \bar{I} represent a vector of all the currents of interest. The representation sought is

$$\bar{I} = \bar{I}_0 + \begin{bmatrix} \frac{\partial I_1}{\partial x_1} & \frac{\partial I_1}{\partial x_2} & \dots & \frac{\partial I_1}{\partial x_n} \\ \frac{\partial I_2}{\partial x_1} & \frac{\partial I_2}{\partial x_2} & \dots & \frac{\partial I_2}{\partial x_n} \\ \vdots & \vdots & \dots & \vdots \\ \frac{\partial I_m}{\partial x_1} & \frac{\partial I_m}{\partial x_2} & \dots & \frac{\partial I_m}{\partial x_n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \equiv \bar{I}_0 + A \bar{x} \quad (12)$$

Every switch setting change delivers information about one column of the matrix in eq. (12). By tracking the changes in the optimization index, the vector \bar{f} is computed

$$\mathfrak{S} = \mathfrak{S}_0 + \begin{bmatrix} \frac{\partial \mathfrak{S}}{\partial x_1} & \frac{\partial \mathfrak{S}}{\partial x_2} & \dots & \frac{\partial \mathfrak{S}}{\partial x_m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \equiv \mathfrak{S}_0 + \bar{f} \cdot \bar{x} \quad (13)$$

The \bar{b} in eq. (11) is

$$\bar{b} = \bar{I}_{s-\text{rated}} - \bar{I}_0 \quad (14)$$

This technique does not avoid the local well problem. However, the algorithm is fast. A Monte Carlo technique involving randomly changing the initial guess has been tested with great profit. The solution time varies linearly with the number of Monte Carlo simulations. For one test problem, the error decreases by a factor of 18 if six Monte Carlo iterations are employed versus one iteration.

5 Conclusions

An approach to reconfiguration is presented which consists of replacing the grid with a subsystem of series and parallel impedances. Using this equivalent representation, optimization equations are listed to maximize power flow through the equivalent parallel impedances and to minimize power flow through the series components. This approach proves useful on a small test grid. Very large systems deserve more research, because a global discrete optimization may become unmanageable in size. Integer optimization delivers only local reconfiguration solutions.

References:

- [1] K. Butler and N. Sarma, "General Reconfiguration Methodology for AC Radial Shipboard Power Systems", *IEEE Power Engineering Society Winter Meeting*, 2000, Vol.: 2, 23-27 Jan. 2000, pp. 1226-1230.
- [2] K. Butler, N. Sarma, and V. Prasad, "A New method of network reconfiguration for service restoration in shipboard power systems", *IEEE Transmission and Distribution Conference*, Vol. 2, 11-16 April 11-16, 1999, pp. 658 - 662.
- [3] S. Lewis and T. Baldwin, "Distribution load flow methods for shipboard power systems", *Industry Applications Conference*, Vol. 2, Oct. 12-16 2003, pp. 812-819.
- [4] Wolsey, Laurence A., *Integer Programming*, John Wiley & Sons, 1998.
- [5] Nemhauser, George L. and Laurence A. Wolsey, *Integer and Combinatorial Optimization*, John Wiley & Sons, 1988.
- [6] Hillier, Frederick S. and Lieberman Gerald J., *Introduction to Operations Research*, McGraw-Hill, 2001.

